Intelligent Systems Project

Team: Ibrahim Alaydi, Ronan Barry, Brian Malone.

Outline:

1. Define 8 – puzzle: The objective of the game is to reorder the numbers until they meet the user’s defined end state. The user also sets the starting state.
2. We’re taking two user inputs (JOptionPane). The *Start state* and the *End state.* We’re creating two arrays to store the states.
3. Validate the input, in the integer range (0-9). Checking for duplicates and removing/outputting error message. Loop, give user a chance to correct their mistake, terminate after 3 attempts. Error at nonnumeric values also.
4. Option between 8 puzzle and 15 puzzle (JOptionPane)
5. Check for exits, dialog box exits, x and cancel.
6. A\* Search algorithm:
   1. Declare three parameters: *Source, Destination, Current* (S, E , C)
   2. Declare a function called *mem(n)* for each single point n.
   3. Initiate two lists called *open* and *closed*.
   4. Initiate two lists call them *open* and *closed*.
   5. Identify all points(m points) that are directly/simply (without any intermediate point) reachable from C: X = {X1, X2,…, Xm}
   6. Those members of X that are not closed are added into the open list.
   7. Add C into closed.
   8. Calculate g(n) for all members of open.
   9. Calculate h(n) for all members of open.
   10. Estimate the value of f(n) for all members of open.
   11. Select a point from open , Oi, where Oi is not an element of closed and f(Oi) is minimum; then assign Oi to C; C:= Oi.
   12. Mem(Oi) = C
   13. Add Oi into closed and remove Oi from open.
   14. If C = E, terminate the algorithm otherwise go back to step d.